You Only Look Once

Abstract

- 今までは classifier と detection を分けてたけど，YOLO はそれを single network で end-to-end でやるよん
- sliding window しないからとても早いよ
- localization の精度では劣るけど，false positives on background は少ないよ

1. Introduction

- 今までは classifier と detection を分けてたけど，YOLO はそれを single network で end-to-end でやった
  - regression problem (回帰問題) にした
- several benefits over traditional method
  i. とても早い
  ii. sliding window と違って全体を見るから context を取れる，その結果 false positives が少ない
  iii. object を抽象化している，実際の写真で学習させたモデルで絵を input として test した時の性能がよい

2. Unified Detection

- input image を $S \times S$ の grid に分割
  - object の中心を含む grid がその object の特定に responsible
- それぞれの grid についてB個の bounding box を定義し，それぞれの box について confidence scores を predict
  - confidence scores reflects
    a. how confident the model is that box contains an object
    b. how accurate it thinks the box is that it predicts
  - confidence = Pr(Object) * $\text{IOU}^{\text{truth}}_{\text{pred}}$
    - IOU: Intersection over Union between the predicted box and the ground truth
      - box 内に object がなかったら 0 で，あったら box と ground truth の重なってる領域の割合

$$\text{IOU}^{\text{truth}}_{\text{pred}} = \frac{\text{pred} \cap \text{truth}}{\text{pred} \cup \text{truth}}$$

      - IOU は class は区別しない，あるかどうかだけ
    - Pr(Class$_i$|Object): その box にobject が存在するとした時，それが class に属する確率
  - bounding box は 5 predictions からなる: x, y, w, h, confidence
    - (x, y): grid の中心
    - confidence: IOU between the predicted box and **any** ground truth box
- bounding box の数 B に関係なく，one set of class probabilities のみ predict
- test 時には class probability と individual box confidence predictions をかける:

$$\text{Pr}(\text{Class}_i|\text{Object}) \times \text{Pr}(\text{Object}) \times \text{IOU}^{\text{truth}}_{\text{pred}} = \text{Pr}(\text{Class}_i) \times \text{IOU}^{\text{truth}}_{\text{pred}}$$

- 最終的な出力は $S \times S \times (B \times 5 + C)$ tensor
  - なぜ C をかけずに足すのか
    - 各 grid は 1 つのクラスを represent するから，つまり $\text{Pr}(\text{Class}_i|\text{Object})$ (class probability map ができる)
  - なぜ $B \times 5$なのか
    - その bounding box の x, y, w, h，及び class を無視したobjectのある確率，つまり IOU
- YOLO on PASCAL VOC を evaluate する時には $S = 7, B = 2, C = 20$

2.1 Network Design

- 24 convolutional layers and 2 fully connected network
  - GoogLeNet に inspire されたけど，inception module (concat のやつかな？) の代わりに 3x3 conv のあとに 1x1

reduction layers
    - 最終的な出力は 1 channel が 1 つの座標もしくは確率に対応，各要素は grid に対応
        - 詳細は論文中の fig.3 に記載
    - alternative 1x1 convolutional layers が何やってるかいまいち掴めないから実装見る
        - chainer 実装だけど以下の雰囲気
        - bias は $Wx + b$ の b だよね？

```
conv5  = L.Convolution2D(64, 128, ksize=3, stride=1, pad=1, nobias=True),
bn5    = L.BatchNormalization(128, use_beta=False, eps=2e-5),
bias5  = L.Bias(shape=(128,)),
conv6  = L.Convolution2D(128, 256, ksize=3, stride=1, pad=1, nobias=True),
bn6    = L.BatchNormalization(256, use_beta=False, eps=2e-5),
bias6  = L.Bias(shape=(256,)),
```

2.2 Training

- Pretrain
    - ImageNet (1000 classes, 224 x 224) で first 20 convs を pretrain
    - 1 ween train して 88% accuracy
- Fine Tuning
    - 448 x 448 にした(detection では fine-grained visual information が求められるため)
    - bounding box の w, h を 0~1 に正規化，x, y は grid の中心に，かつ 0~1 にした
- Activation
    - Final layer: linear activation function
    - Other layers: leaky ReLU ($x < 0$ で 0 じゃなくて $0.1x$ を返す)
- Loss Function と問題点
    - sum-squared error (2乗和？)
    - optimize しやすいから，でも性能少し劣る
    - may not be ideal な classification error と localization error を同等に扱っている
    - 多くの grid には object は存在せず，その時 confidence score は zero にしたいが，これは object を含む cell による影響を上回る
        - この欠点によりモデルは不安定になる
    - 大きな box と小さな box の error を同一視
        - 大きい box 内の小さなズレは小さい box におけるそれによる loss よりも小さくしたい
- 解決策 (remedy)
    - bounding box の座標に関する loss を大きくする ($\lambda_{\text{coord}} = 5$)
    - object を含まない grid の confidence による loss を小さくする ($\lambda_{\text{noobj}} = 5$)
    - box の平方根を予測するようにする
- どの bounding box を選ぶか
    - the highest IOU
- loss function
    - https://www.slideshare.net/ssuser07aa33/introduction-to-yolo-detection-model に詳しい
- params
    - 135 epochs on the training and validation data from PASCAL VOC 2007 and 2012
    - batchsize = 64
    - momentum = 0.9
    - decay = 0.0005
- learning rate
    - first epoch: 10^{-3} から 10^{-2} までraise
    - 次の 75 epochs は 10^{-2}，そこから 30 epochs は 10^{-3}，最後の 30 epochs は 10^{-4}
- to avoid overfitting
    - dropout
        - first FC の後に 0.5 の dropout layer
    - data augumentation
        - random scaling and translations of up to 20%
        - randomly adjust the exposure (露光) and saturation (彩度) を HSV color space で 1.5 倍の範囲で変化

2.3 Inference (推測)

- 大きな object や border of multiple cells の近くの object は複数の cell に well localized され得る
  - Non-maximal suppression を用いて重複して detect されることを防いだ
  - R-CNN, DPM ではいまいちだったが，この導入により mAP が 2~3% 上昇

2.4 Limitations of YOLO

- one object per one grid cell なので，近くに複数物体がある場合に弱い
  - 羊や鳥の群れのようなグループでいるものに弱い
  - これ魚やばくね？
- generalize しようとするのでざっくりした (coarse) detection
- box の大きさと loss の問題
  - small error in a small box は IOU に多大な影響を及ぼす
    - これも平方根取ればいいのに

3. Comparison to Other Detection Systems

- 他の手法との比較

4. Experiments

- R-CNN と YOLO が犯したミスを比較し，その傾向を分析
  - Fast R-CNN の score を修正できる
  - false positive が少ない

5. Conclusion

- まとめてた